Robust Sensorimotor Representation to Physical Interaction Changes in Humanoid Motion Learning

Toshihiko Shimizu, Ryo Saegusa, *Member, IEEE,* Shuhei Ikemoto, Hiroshi Ishiguro, *Member, IEEE,* Giorgio Metta, *Senior Member, IEEE,*

***Abstract*—This study proposes a learning from demonstration (LfD) system based on a motion feature, called phase transfer sequence. The system aims to synthesize the knowledge on humanoid whole body motions learned during teacher-supported interactions, and apply this knowledge during different physical interactions between a robot and its surroundings. The phase transfer sequence represents the temporal order of the changing points in multiple time sequences. It encodes the dynamical aspects of the sequences so as to absorb the gaps in timing and amplitude derived from interaction changes. The phase transfer sequence was evaluated in reinforcement learning of sitting-up and walking motions conducted by a real humanoid robot and a compatible simulator. In both tasks, the robotic motions were less dependent on physical interactions when learned by the proposed feature than by conventional similarity measurements. Phase transfer sequence also enhanced the convergence speed of motion learning. Our proposed feature is original primarily because it absorbs the gaps caused by changes of the originally acquired physical interactions, thereby enhancing the learning speed in subsequent interactions.**

***Index Terms*—change detection, dimensionality reduction, learning from demonstration, physical human-robot interaction**

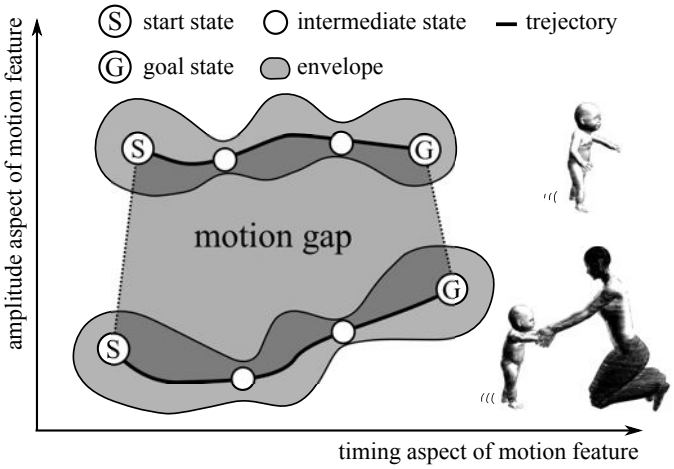


Fig. 1. Conceptual figure that illustrates a gap between two whole body motions in different physical interactions. The area bounded by two motor trajectories is termed a motion gap that shows differences of timing and amplitude in motions. The envelope shows the boundary of the trajectories that lead to a successful task achievement. The states near the shrinking of the envelope (bottlenecks) are important for achieving the motions.

I. Introduction

INFANTS acquire the motor skills necessary for self-sustained walking partly by social motor interactions [1] [2] However, how infants extract skills from their experiences of teacher-supported (TS) walking and apply them to self-sustained (SS) walking at an early age is less clearly understood. Although TS and SS walking are outwardly similar, the internal motor controls are physically different because teacher interaction imposes external forces and spatial constraints during locomotion. Since motion must be initially achieved for subsequent adaptation, such as optimizing the acquired motion, we focus on accelerating the acquisition of a motion with identical purpose, but different physical interactions using knowledge gained during a TS interaction.

In robotics, motor skills can be learned from teacher interactions by a method known as learning from demonstration (LfD), which widely employs dynamic motion primitives (DMP) [3] to govern whole body motions of humanoid robots. In the DMP framework, human motions are represented by a set of differential equations whose parameters are calculated by locally weighted regression. Under this system, a robot learned how to swing a tennis racket. To mimic bipedal walking, DMP employs a central pattern generator (CPG) [4] that uses the walking motions of a human. Robots can also be made to perform traditional Japanese dances by segmenting their motion into key postures where the velocities of the end-effector become zero [5]. The invariant features that achieve dynamic roll-and-rise (RAR) in human motion have been evaluated, and reproduced in a humanoid robot by passing the bottleneck states [6]. In a study of mutual adaptation during physical human-robot interactions [7], represented by the Gaussian mixture model, a whole-bodied pneumatic actuated robot acquired standing up and walking motions with human support. A standing-up motion from a chair on an inclined slope is generated by the motion phase decision tree algorithm [8], in which a set of successful and failed trials is

T. Shimizu is with the Department of Mechanical Engineering, Kobe City College of Technology, 8-3, Gakuen-higasi-machi, Nishi-ku, Kobe, Hyogo, 651-2194, JAPAN. (e-mail: ts8@kobe-kosen.ac.jp)

R. Saegusa is with the Center for Human-Robot Symbiosis Research, Toyohashi University of Technology, 1-1 Hibarigaoka, Tempaku, Toyohashi, Aichi, 441-8580, Japan. (e-mail: ryos@ieee.org)

S. Ikemoto is with the Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University, E6-411, 2-1, Yamada-oka, Suita, Osaka, 565-0871, Japan. (e-mail: ikemoto@ist.osaka-u.ac.jp).

H. Ishiguro is with the Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, 1-3, Machikaneyama, Toyonaka, Osaka, 560-8531, Japan. (e-mail: ishiguro@sys.es.osaka-u.ac.jp).

G. Metta is with the iCub Facility and the Robotics, Brain, and Cognitive Sciences Department, Istituto Italiano di Tecnologia, Via Morego 30, 16163, Genova, Italy. (e-mail: giorgio.metta@iit.it)

This work was carried out at the Robotics, Brain and Cognitive Sciences Department, Istituto Italiano di Tecnologia.

Manuscript received May 7, 2012; revised Dec 7, 2013.

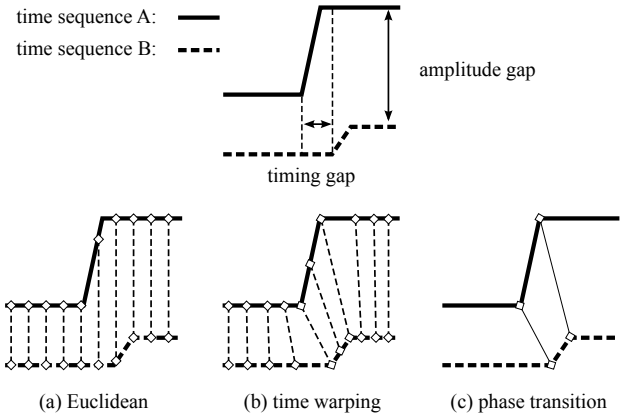


Fig. 2. Conceptual illustrations of three similarity measurements between two time sequences with the gaps in the timing and amplitude. Euclidean distance is affected by both gaps, while time warping is affected only by the amplitude gap. Phase transition is not affected by any of the gaps, because the two time sequences have two phase transitions in common.

built into a decision tree of motion phases.

The trajectories of the joint angles is represented by Gaussian mixture regression (GMR) [9], which enables robot manipulations to be reproduced in different positions. Parametric hidden Markov models (PHMMs) [10] represent the trajectories of the joint angles together with their effect on object motions, obtained from visually observed human demonstrations. This approach achieves action recognition and reproduction of manipulative tasks; however, neither GMR nor PHMMs have been examined in a whole-bodied motion generation setup.

These methods reproduce the demonstrated motions of robots working in different conditions, such as joint angle mapping between a human and a robot, different object positions, and slope inclination. However, for motion reproduction, each physical interaction requires its own representation (Fig. 1), because the motor information is characterized by raw time sequences (e.g., joint angle or Cartesian space) and their Euclidean distance (ED). In Fig. 1, the successful region of the motion information (in terms of amplitude and timing) is delineated by the envelope, and is separate for TS and SS walking. The gap between the two successful regions in Fig. 1 is fundamentally derived from the gap in amplitude and timing of the two time sequences, as shown in Fig. 2. In terms of the similarity measurement between the two time sequences, the ED is affected by both the amplitude and the timing gap. Therefore, ED-based representation is expected to be limited in each physical interaction. In representing sequential information, the timing dependence is often relaxed by applying time warping (TW), although the amplitude gap remains in the representation.

Our current research focuses on phase transitions, which represent points of change between the time sequence dynamics (Fig. 2(c)). Phase transitions should more effectively limit the amplitude and timing gaps than ED and TW, because they represent only the number of events in the time sequence, regardless of the size of the amplitude and timing gaps. The time sequence dynamics can vary, for example, they can be smooth, sudden, or noisy. To obtain phase transitions between these dynamics, we employed a change detection method called singular spectrum transformation (SST [11]), which evaluates the variation in the time sequence at the center of a sampling window. By feeding the time sequence to the SST at each time step, the changes are sequentially detected.

We assume that if different physical interactions are performed for the same purpose, the temporal orders of the phase transitions of the motions are similar. Thus, we propose "phase transfer sequence" (PTS). This motor representation converts motion phase transitions into symbols by sorting the multiple phase transitions of the time sequences in temporal order. Since phase transitions can limit the timing and amplitude gaps caused by changes in interactions, PTS is expected to efficiently handle differences in the motions. Moreover, PTS can potentially encode important states for motion achievement, because each phase transition is probably caused by meaningful events, such as stepping on the ground and twisting at the waist. The similarity between two PTSs was computed by the longest common subsequence (LCS) [12].

We then applied the PTS to a robotic LfD system, and verified that the representation is robust in different physical interactions. It also accelerates the learning convergence speed in simulations and experiments with a real humanoid robot. In the experiments, the enhanced learning speed in the SS interaction, given the knowledge acquired during the TS interaction, was verified by walking and sitting-up motions.

This paper is organized as follows: Section II describes an LfD system based on our new representation, "phase transfer sequence". The proposed feature is experimentally compared with other similarity measurements in Section III. Section IV discusses the proposed system and concludes the study.

II. Learning from demonstration with phase transfer sequence

We apply PTS to an LfD system, aiming to accelerate the acquisition of motions in physical interactions that differ from the TS interactions, but which serve the same purpose. The temporal order of the phase transitions in different interactions is assumed similar if the motions seek the same goal. Figure 3 shows that the proposed process consists of three phases: demonstration, representation, and reproduction.

A. Demonstration phase

A robot is instructed by the teacher to execute a task N_{dm} times, and observes the time sequences of multiple internal sensors, as shown in Fig. 3(A). Note that time sequences are assumed to be derived from internal sensors on the robot platform. The robot is provided with no additional knowledge of the task motion.

B. Representation phase

Figure 3(B) shows an overview of the three phases in the proposed representation; namely, binarization, symbolization, and selection. In binarization, each time sequence is fed into the system (i) and transformed into a binary sequence, where 1 corresponds to a phase transition (ii). In the symbolization

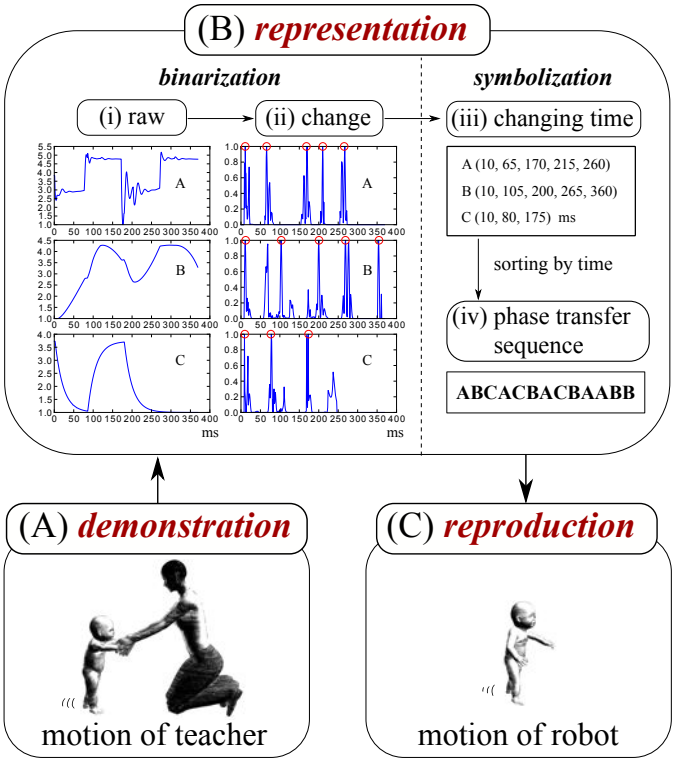


Fig. 3. The proposed system outline. The process consists of three phases: demonstration, representation and reproduction. The concept of PTS is shown in (B) representation. The raw data (i) are transformed into change scores (ii). Then the peaks of each change score are collected (iii) and the PTS is obtained by sorting the sensor symbols by time (iv). The symbols of each sensor are denoted as **A**,**B**,**C**. In the demonstration phase, the robot records the sensory sequences using its sensors, then the sequences are converted into PTS. In the reproduction phase, the PTS obtained in the TS interaction is used as a guide for enhancing the learning convergence in new physical interactions.

phase, the changing times of each sequence (associated with a different symbol) are identified (iii) and the symbols are concatenated in the temporal order of transition occurrence, generating a sequence (iv). The PTS is generated by algorithm 1. Once the PTS of each demonstration has been computed, the selection phase identifies a reference trial to pass to the reproduction phase.

The proposed symbolic representation reduces the dimensionality of the system for multiple time sequences. Its symbolic nature is compatible with traditional string computation algorithms, such as the LCS [12]. By virtue of SST, our representation is also applicable to multiple robots equipped with multimodal sensors.

1) Binarization: To extract phase transitions from time sequences with varying dynamics (such as smooth, sudden, and noisy), we adopted SST [11] as the change detection method. The SST evaluates the variation in the time sequence at the center of a sampling window. Since SST is based on singular value decomposition (SVD), which is applicable to any matrix, it can be applied to various time sequences without requiring ad-hoc tuning.

This research uses robust singular spectrum transformation (RSST) [13], an improved version of SST that reduces the number of parameters to only two; the window size n_r and the number of windows n_c. The computational procedure of RSST is described in algorithm 2. A change score sequence is computed from the time sequences fed into the RSST at every time step. The phase transitions are sequentially detected as peaks in the change score sequence, as shown in Fig. 3(B)(ii). The change score sequences are then binarized by setting 1 if a peak is present and 0 otherwise.

2) Symbolization: The symbolization procedure collects the times at which the score changes to 1 in each change score sequence (iii). Each time is labeled with a symbol corresponding to the sensor identifier, and the symbols are concatenated in temporal order (iv). For example, if a sensor with changing times $(1, 4)$ is symbolized by **A**, and another sensor with changing times $(2, 3)$ is symbolized by **B**, the PTS is **ABBA**. The sequence of symbols in the PTS is denoted $\mathbf{o} = o[1, ..., n_{\mathbf{o}}]$, where $o[i]$ is the ith symbol of $\mathbf{o}$, and $n_{\mathbf{o}}$ is the number of symbols. Note that the same symbol can appear consecutively multiple times, as in the previous example.

3) Selection: The PTS $\mathbf{o}_r$ of the i^rth demonstration is used as a reference during the reproduction phase. The reference index i^r is selected by the following heuristic:

$$i^r = \arg\max_{i \in \mathbf{N}_{dm}} \sum_{j \in \mathbf{N}_{dm} \setminus \{i\}} |LCS(\mathbf{o}_i, \mathbf{o}_j)| \qquad (1)$$

where $\mathbf{N}_{dm} = \{1, ..., N_{dm}\}$ is the set of the demonstration indices, N_{dm} is the number of demonstrations, $LCS(\mathbf{o}_i, \mathbf{o}_j)$ is the longest common subsequence between $\mathbf{o}_i$ and $\mathbf{o}_j$, and $|LCS(\mathbf{o}_i, \mathbf{o}_j)|$ is its length.

The length of an LCS is calculated as follows:

$$|LCS(\mathbf{o}_i, \mathbf{o}_j)| = L[n_{\mathbf{o}_i}, n_{\mathbf{o}_j}], \qquad (2)$$

where $L[\tilde{i}, \tilde{j}]$ $(0 \leq \tilde{i} \leq n_{\mathbf{o}_i},\ 0 \leq \tilde{j} \leq n_{\mathbf{o}_j})$ is a score matrix built by dynamic programming as follows:

$$L[\tilde{i}, \tilde{j}] = \begin{cases} 0 & (\tilde{i} = 0, \tilde{j} = 0), \\ L[\tilde{i} - 1, \tilde{j} - 1] + 1 & (o_i[\tilde{i}] = o_j[\tilde{j}]), \\ max\{L[\tilde{i}, \tilde{j} - 1], L[\tilde{i} - 1, \tilde{j}]\} & (otherwise). \end{cases} \qquad (3)$$

C. Reproduction phase

In the reproduction phase, the reference PTS $\mathbf{o}_r$ provides a guide post for motor exploration in new physical interactions, as shown in Fig. 3(C). In this phase, the system autonomously chooses actions and uses its previously acquired knowledge to adapt the demonstrated motion to different interactions.

The new motions are learned by reinforcement learning (RL). In the RL framework, an agent performs actions until a goal state is achieved, as shown in Fig. 4. When it reaches a valuable goal state, the agent receives rewards from the environment. If the reward is given only in the final goal state, all trials with intermediate failures are wasted, and learning convergences requires a vast number of trials.

In this research, motor exploration is guided by PTS, which provides agents with sub-goal rewards at intermediate states if the similarity between the current PTS $\mathbf{o}_c$ and the reference PTS $\mathbf{o}_r$ is high. The longer the common subsequence, the more similar the phase transitions in the interactions.

Note that PTS can be generated both offline and online. We selected the offline approach. At the end of each *episode*, the

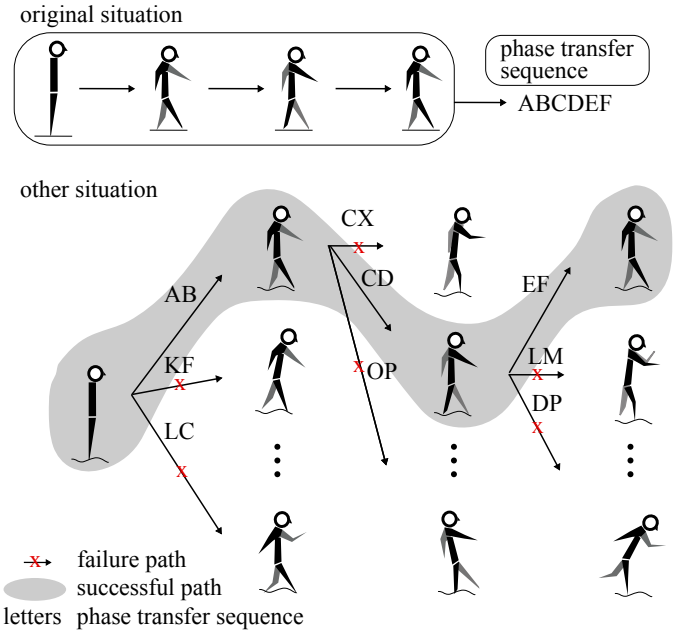


Fig. 4. A schematic view of motion learning. Phase transfer sequence obtained from the original situation is reused as a guide for realizing successful motions in other situations. Since the path with a similar phase transfer sequence to the original one is reinforced by the sub-goal rewards at intermediate states, the learning is potentially enhanced.

PTS is generated and the action-value function is updated. In the online approach, the PTS can be progressively generated by adding a symbol as each new transition is identified. At this moment, a reward is computed from the LCS between the PTS and the reference PTS.

1) Q-learning: We employed Q-learning [14] for motion learning. The action-value function is defined as follows:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a)+\alpha(r(s,\acute{s},t)+\lambda \max_{\acute{a}\in A} Q(\acute{s},\acute{a})) \quad (4)$$

where $s \in S$ and $a \in A$ are the current state and the action, respectively, and S and A are the sets of states and actions, respectively. $\acute{s}$ and $\acute{a}$ are the next state and the next action, respectively. t denotes the time at which the agent reaches $\acute{s}$. In this research, the action is defined as the whole body motion from the current posture to the next posture. These postures are interpolated assuming a bell-shaped velocity profile. The states are the indices of discretized information, namely, the grid index of the workspace and the index of the reference vector. Actions are selected by an ϵ greedy strategy. In this study, *episode* denotes a trial consisting of the state transitions in the selected actions; a *run* consists of multiple *episodes*. The following parameters are constant: $\alpha = 0.25$, $\lambda = 0.9$, and $\epsilon = 0.5$. The reward function without sub-goal rewards is defined as

$$r(s,\acute{s},t) = \begin{cases} 1 & (s \in S \text{ and } \acute{s} = s_g \text{ and } t = t_e), \\ -1 & (fail), \\ 0 & (\text{else}) \end{cases} \quad (5)$$

where s_g is the goal state, t_e is the end time of the experiment, and $fail$ denotes task failure.

The reward function with sub-goal rewards is defined as

$$r'(s,\acute{s},t) = \begin{cases} 1+f(s,\acute{s}) & (s \in S \text{ and } \acute{s} = s_g \text{ and } t = t_e), \\ -1 & (fail), \\ f(s,\acute{s}) & (\text{else}), \end{cases} \quad (6)$$

$$f(s,\acute{s}) = \begin{cases} 0 & (s = s_s), \\ g(\acute{s}) - g(s) & (\text{else}), \end{cases} \quad (7)$$

$$g(s) = \begin{cases} g^E(s) & (\text{with ED}), \\ g^T(s) & (\text{with TW}), \\ g^P(s) & (\text{with PTS}) \end{cases} \quad (8)$$

where s_s is the start state, $f(s,\acute{s})$ is the sub-goal reward obtained from s to $\acute{s}$, and $g(s)$ is the sub-goal reward obtained when the agent reaches s. The sub-goal rewards $g^E(s)$, $g^T(s)$, and $g^P(s)$ will be described later in this section. To enable the agent to evaluate the sub-goal reward obtained from s to $\acute{s}$, the current sub-goal reward $g(s)$ is subtracted from the next sub-goal reward $g(\acute{s})$ according to (7) (except when $s = s_s$).

2) Similarity Measurements: Three types of sub-goal rewards, ED, TW, and PTS, are compared in our experiments. We assume a set of sensors $P = \{p_1, ..., p_{N_p}\}$, where N_p denotes the number of used sensors. The sensor time sequences from the start time t_s to time t are defined as $X(t) = \{\mathbf{x}_1(t), ..., \mathbf{x}_{N_p}(t)\}$, where $\mathbf{x}_i(t) = \{x_i(t_s), ..., x_i(t)\}$ is the time sequence for sensor p_i.

The first index based on the ED is defined as follows:

$$g^E(s) = \frac{1}{1 + c^E \sum_{i=1}^{N_p} \|\mathbf{x}_i^c(t) - \mathbf{x}_i^r(t)\|}, \quad (9)$$

where t is the current time (at which the agent reaches s), $\mathbf{x}_i^c(t)$ and $\mathbf{x}_i^r(t)$ are the time sequences of the current trial and the i^r trial, respectively, and c^E is a constant that scales the sum of the EDs. Figure 2(a) shows that this function compares the time sequences of the current and the i^r trials from the start time t_s until the current time t.

The second index based on TW is defined as follows:

$$g^T(s) = \frac{1}{1 + c^T \sum_{i=1}^{N_p} TW(\mathbf{x}_i^c(t), \mathbf{x}_i^r(t_e^r))}, \quad (10)$$

where the function TW represents the TW distance between $\mathbf{x}_i^c(t)$ and $\mathbf{x}_i^r(t_e^r)$. Here, t_e^r is the end time of i^r, because TW can evaluate two sequences of different temporal lengths. Thus, this function compares the time sequences of the current trial from the start time t_s to the current time t and those of i^r from t_s to t_e^r, as shown in Fig. 2(b).

The last index based on PTS is defined as follows:

$$g^P(s) = \frac{|LCS(\mathbf{o}_r, \mathbf{o}_c(t))|}{|\mathbf{o}_r|}, \quad (11)$$

where $LCS(\mathbf{o}_r, \mathbf{o}_c(t))$ is the LCS of $\mathbf{o}_r$ and $\mathbf{o}_c(t)$. $\mathbf{o}_c(t)$ represents the PTS of $X(t)$ in the current trial. $|\mathbf{o}_r|$ is the length of the reference PTS. If the motion is similar to the i^r motion, all three of the above indices are close to 1. The similarity measurement is inversely proportional to the motion gap (Fig. 2); small motion gap implies high similarity. These indices provide a positive sub-goal reward expressing the extent to which the motion is similar to the reference demonstration.

III. EXPERIMENT

The proposed method was evaluated on a real humanoid robot and its corresponding simulator. The robot platform was iCub [15], a child-scale full-body humanoid robot (approximately 104 cm tall) with 53 degrees of freedom (DOF). The robot platform was controlled with CPU clusters via YARP [16]. Force/torque sensors were mounted on the four limbs, and an inertial sensor was mounted on the head, joint encoders, and the corresponding motors of all DOFs. The simulated robot corresponds to a real robot platform.

The selected tasks were the discrete, rhythmic motions of sitting-up and walking. First, the robot received human assistance to complete task motions (TS interaction). It then executed RL without assistance (SS interaction). The first objective of the experiments was to investigate how knowledge extracted from demonstrations in the TS interaction improves the learning convergence in the SS interaction. The TS and SS motions were different because the physical interactions were inherently different. Experiments showed that SS motions cannot be directly achieved from TS motions. The second objective was to compare the robustness of the time sequence representation under ED, TW, and the proposed PTS. The PTS configurations in the sit-up and walk experiment are listed in Table I. The asymmetries in the RSST parameters reflect asymmetries in sensor input and in other conditions, such as body mass distribution, joint speeds during movement, joint friction, and contact with the environment. These parameters were empirically determined by tuning them as follows: First, the sensor data sequences were acquired from the TS motion. Next, all RSST parameters were varied, and the time sequences and their change scores were visualized. The parameters were then tuned to detect changes in the TS motions.

In the sitting-up experiment, the robot learned sit-up motions through being lifted by the teacher. To ease the lifting task for the teacher, the pitch joints of both shoulders and hips of the robot were freed, and the pitch joint of the torso was controlled by a torque-based control. The other joints were retained in their home position by position-based control. Initially, the robot was laid on the ground, as shown in Fig. 5 left. The teacher lifted the robot by holding both its lower arms.

TABLE I
CONFIGURATIONS FOR THE PHASE TRANSFER SEQUENCE.

Configurations for sit-up and roll-and-rise experiment.				
symbol	part name	sensor name	RSST (real/simulation)	
			n_r	n_c
0	left arm	force x	8 / 8	8 / 8
1	right arm	force x	6 / 6	6 / 6
2	left leg	force x	8 / 8	8 / 8
3	right leg	force x	8 / 8	8 / 6
4	left leg	hip pitch encoder	10 / 10	4 / 10
5	right leg	hip pitch encoder	10 / 10	10 / 10
6	left arm	shoulder pitch encoder	10 / 10	4 / 10
7	right arm	shoulder pitch encoder	10 / 10	10 / 10
8	torso	torso pitch encoder	10 / 10	10 / 10
a	head	inertial pitch	8 / 8	4 / 4
b	head	inertial gyro y	16 / 16	4 / 4
Configurations for walk experiment.				
symbol	part name	sensor name	RSST (real/simulation)	
			n_r	n_c
0	torso	torso roll encoder	10 / 8	10 / 6
1	right leg	hip pitch encoder	20 / 8	10 / 6
2	left leg	hip pitch encoder	16 / 8	16 / 6
3	right leg	hip roll encoder	16 / 8	16 / 6
4	left leg	hip roll encoder	14 / 8	14 / 6
5	right leg	force z	8 / 8	8 / 8
6	left leg	force z	8 / 8	8 / 8
The axes of sensors are corresponding with those in Fig.9.				

Fig. 5. Screenshots of the teaching experiment of the sit-up motion.

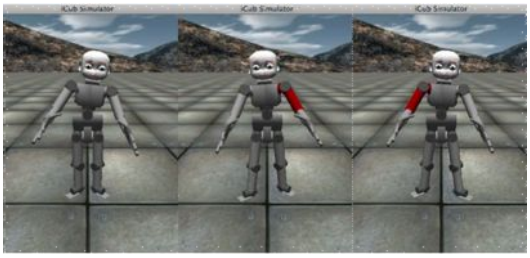


Fig. 6. The initial posture and the implemented reactions. The left figure shows the initial posture, the center figure is the reaction to a stimulus on the left arm, and the right figure is the reaction to a stimulus on the right arm.

In the walking experiment, teachers gripped the robot's arms and taught it walking motions. To ensure safe human-robot interactions, we implemented the following reactive motion: when the teacher elevates one arm of the robot, the robot detects the stimulus, steps forward with the opposite-side leg, and bends the torso to the opposite side, as shown in Fig. 6. The joint angle configurations in the walking reactions are listed in Table II. The initial posture and right and left walking reactions (denoted "home," "right," and "left" in Table II) are shown in the left, center, and right sections of Fig. 6, respectively. The hip yaw joints of both legs were externally rotated to gain stability and propulsion [17]. Both arms were extended forward to interact with the teacher.

The actions and reactions of the robot are defined as the whole body motion from the current posture $\Theta = \{\theta_1, ..., \theta_{N_D}\}$ to the next posture $\Theta^d = \{\theta_1^d, ..., \theta_{N_D}^d\}$, where θ_i is the ith joint angle and N_D is the number of whole body joints. Each joint trajectory from the current joint angle θ_i to the next joint angle θ_i^d is generated by a minimum jerk trajectory generator. This generator produces a smooth interpolation between θ_i and θ_i^d assuming a bell-shaped velocity profile (maximum velocity $\tilde{v}$). This trajectory is then followed using position-based control with a PID controller.

A. Learning and reproduction

The robot simulator was implemented via an interactive GUI. The mouse drag-and-drop movement was translated into the amplitude of the external impulsive force to be applied to the robot arms. Forces were added to the right and left

TABLE II
ACTION DEFINITIONS WITH JOINT ANGLE.

joint name.	walk reactions			walk actions				sit-up actions			roll-and-rise actions						
	home	right	left	init	0	1	2	0	1	2 (keep)	0	1	2	3	4	5	6
torso roll	8	-8	8	8	-8	8	-8	0			0						
torso pitch	0			0				0	8	0 or 8	0		48	0	54	48	24
left hip pitch	0	-2.4	8	0	-2.4	19.8	-2.4	0	88	0 or 88	0		88	56	85	85	85
right hip pitch	0	8	-2.4	0	11	-2.4	25.6	0	88	0 or 88	0		88	56	85	85	85
both hip roll	0	8		0	8			0			0						
both hip yaw	48			48				0			0						
both knee pitch	0			0				0			0			120			
both ankle pitch	0			0				0			0				-20		
both shoulder pitch	-30			0				0	-88	0 or -88	0	-88	-88	-40	-40	-80	-88
both shoulder roll	30			30				30			30	45					
both elbow pitch	45			0				45			45	15.5			40	15.5	
other joints	0 (All angles above are given in degree $^\circ$)										Both hands are close and head pitch is -24.						
(joint velocity $\tilde{v}$)	(max 30 $^\circ/s$)							(max 20 $^\circ/s$)			(max 60 $^\circ/s$)						

arms by clicking the left and right mouse buttons, respectively. The forces could be simultaneously applied to both arms. The sensor information was sampled from 0.1 s before the interaction to 0.1 s after the interaction at a sampling frequency of 200 Hz. The interaction session in the sitting-up experiment was complete when the robot had achieved the sitting posture. In the walking experiment, it was complete when the robot had undertaken a specified number of successful steps. The TS motions were taught in N_{dm} trials, and i^r was selected by (1). Because of the limitation of the real robot, learning SS walk with the real robot was not conducted.

1) Learning of self-sustained sit-up in simulation: The sitting-up experiment involved three actions $A_s = \{a_0, a_1, a_2\}$ and three states $S_s = \{s_0, s_1, s_2\}$. The actions are listed in Table II. The action a_2, called "keep", maintains the current posture a_0 or a_1. Decisions are made at time $\{0.0, 0.6\}$ s, and one $episode$ ends at time $t_e = 1.2$ s. The state is defined as the index of the nearest reference vector to the current state vector as follows:

$$s_i = \underset{j\in\{0,1,2\}}{\arg\min} |\mathbf{s} - \mathbf{s}_j^r| \tag{12}$$

where $\mathbf{s} = (\Phi, \Theta)$ and $\mathbf{s}_j^r = (\Phi_j, \Theta_j)$ are the current state and reference vectors, respectively, Φ is the Euler angles of the head, and Θ is the joint angles of the shoulder and the hip pitch encoders of the limbs. The reference vectors $\{\mathbf{s}_0^r,\ \mathbf{s}_1^r,\ \mathbf{s}_2^r\}$ were recorded at time $\{0.0, 0.6, 1.2\}$ s by performing the action sequences $\{a_1, a_1\}$ at time $\{0.0, 0.6\}$ s. These successful sit-up actions are called the initial, intermediate, and final reference, respectively, as shown in Fig. 7. The initial state is $s_s = 0$ and the goal state is $s_g = 2$.

The $episode$ was regarded as successful if the robot reached s_g in 1.2 s; otherwise, it was regarded as failed. If three consecutive $episodes$ have successfully executed, a run has been completed. If the number of $episodes$ exceeds 40, the run is considered failed and terminates. 10 $runs$ are conducted for each reward function. The robot was taught sit-up motions during ten sessions, as shown in Fig. 5.

Figure 7 shows the screenshots of a learned sitting-up motion in the SS interaction. The average reward and the average number of $episodes$ for learning convergence are profiled in Fig. 8(a). The reward without sub-goal rewards

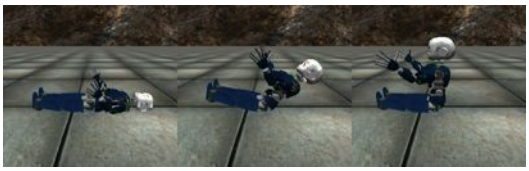


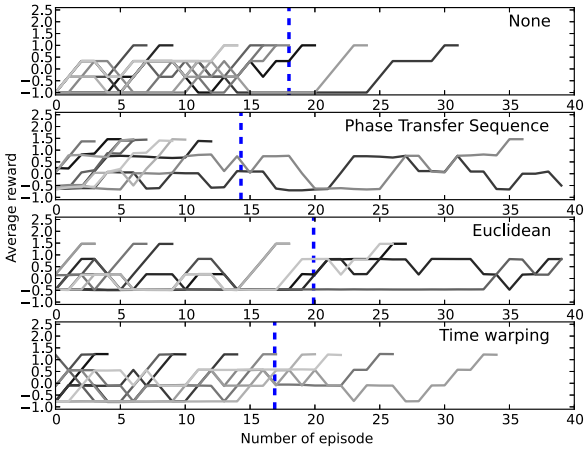
Fig. 7. Screenshots of successful sit-up motion in the SS interaction. The left, middle, and right figures show the initial, intermediate, and final reference of the sit-up, respectively .

represents the base-line learning of the SS sit-up task, and the other cases represent knowledge-based learning through the TS sitting-up experience. The results are summarized in Table III. As expected from the conceptual differences shown in Fig. 2, the convergence speed was improved on average by PTS, TW, and ED, in this order. The ED-based learning result was poorer than base-line, because ED is affected by both the timing and amplitude gap. TW-based learning was improved over base-line learning because it corrects the timing gap, as described in Fig. 2. PTS-based learning achieved the most rapid convergence among the tested procedures, because the phase transitions are unaffected either gap. In PTS representation, 8 $runs$ converged within 14 $episodes$. In the case of PTS and ED, however, at least one run learning failed to converge, because the sub-goal rewards guided the learning toward an incorrect local solution, where it became trapped.

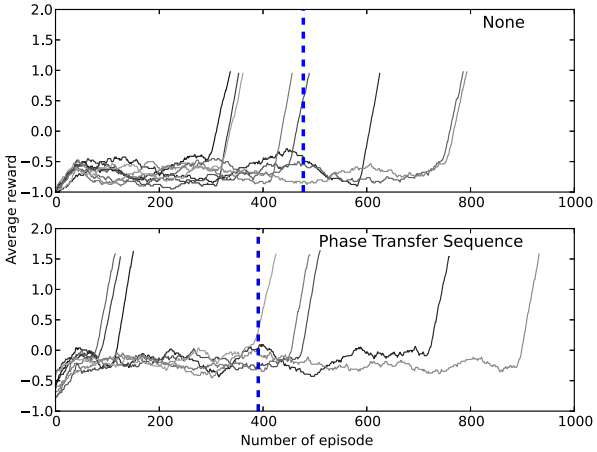
2) Learning of self-sustained walk in simulation: The learning of SS walking involves three actions $A_w = \{a_0, a_1, a_2\}$ and 24 states $S_w = \{s_0, ..., s_{23}\}$. As shown in Fig. 9, the state is defined as the grid index containing the robot head as follows: $s_i = (g_x^i, g_y^i, g_z^i) = (\frac{i}{d_y d_z} \bmod d_x,\ \frac{i}{d_z} \bmod d_y,\ i \bmod d_z)$ $(i = 0, ..., 23)$, where (g_x^i, g_y^i, g_z^i) is the partition index of each axis, and $(d_x, d_y, d_z) =$

TABLE III
LEARNING RESULTS OF SIT-UP EXPERIMENTS.

sub-goal reward	average speed	successful run	under 14 $episodes$
None	18.0	10	2
PTS	14.3	9	8
Euclidean	19.9	8	4
TW	16.9	10	4



(a) The learning result of the sit-up experiment.



(b) The learning result of the walk experiment.

Fig. 8. The average reward acquired during each run. The blue vertical dotted-line shows the average number of *episode* necessary for the learning convergence. In (a), the top plot shows the result without the sub-goal reward, and the second, third and forth plot show the results obtained with that of PTS, ED and TW, respectively. Each line shows the histories of the average reward in a single run, smoothed by a low-pass filter with a 3 *episodes* sampling window. In (b), the upper plot shows the experiments without PTS in the reward function, and the lower plot shows the experiment with PTS. A low-pass filter with an 80 *episode* was used for smoothing. Experiments converged in less than 80 *episodes* and the non-converged runs were excluded from the plot.

$(4, 3, 2)$ is the number of partitions of the work space. The range of the work space is $(lim_x, lim_y, lim_z) = ([-0.1, 0.5], [-0.25, 0.25], [0.5, 1.0])$ m. The initial and goal state are $s_s = 3$ and $s_g = 19$, respectively. The actions are listed in Table II. Action selection is decided at times $\{0.0, 0.4, 0.8\}$ s, and one *episode* ends at time $t_e = 1.2$ s. The initial posture of the robot at t_0 is set at the home position (denoted "init" in Table II).

When the agent reaches the goal state, it receives a positive reward from the reward function. When the agent exits the area (ranges beyond the states), it receives a negative reward from the reward function; namely, $fail$. Each trial lasts 1.6 s or until the robot reaches the goal state s_g, whichever is the smaller. If the robot fails while executing a motion, the trial

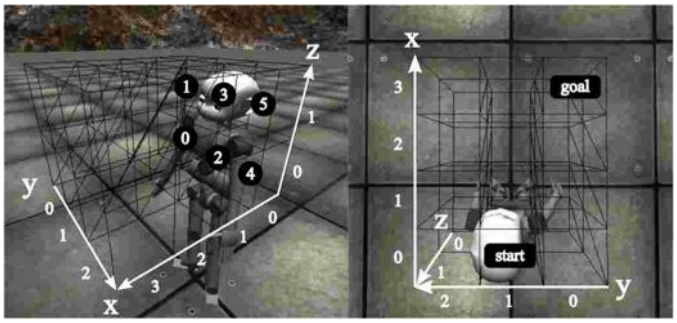


Fig. 9. State definitions. The state is defined as the grid index which divides the work space. In the left figure the black nodes with a white number show the indices of the states. In the right figure the black nodes with white letter show the initial state (denoted by "start") and the goal state (denoted by "goal").

(a) A walk motion instructed by a teacher with an interactive GUI.

(b) A walk motion generated by the robot after learning.

(c) A badly instructed walk motion. At the third step, the right leg is slightly stuck to the ground (shown in 4th figure from the left).

Fig. 10. Screenshots for walk experiments in simulation.

is also terminated. One run is organized as follows: if three consecutive *episodes* are successful, the run is completed, but if the number of *episodes* exceeds 1000, the run is regarded as failed. To compare the learning speed with PTS rewards and with no sub-goal rewards, 10 $runs$ of each algorithm were performed. The robot was taught walk motions during ten sessions, as shown in Fig. 10(a).

Figures 8(b) and 10(b) show the profiles of the average reward and the obtained walk motion, respectively. The PTS rewards yielded 9 successful runs. Conversely, when no sub-goal rewards were received, 10 run were successful. The average convergence speed with and without the PTS rewards was 390 *episodes* and 477 *episodes*, respectively. Within 200 *episodes*, 4 $runs$ and 1 run converged, respectively. Thus, the PTS effectively guided the learning speed enhancement, despite the altered physical interaction conditions. The non-converged PTS case was possibly caused by trapping in a local minimum during learning, or by noise in the $\mathbf{o}_r$, with unnecessary subsequent effects on SS walking.

(a) Interaction for teaching a sit-up motion.

(b) Reproduction of the learned sit-up motion.

Fig. 11. Sit-up experiments by the real robot.

3) Learning self-sustained sit-up with a real robot: The proposed method was verified in sit-up experiments performed on a real robot. Robot safety was ensured by laying the robot on a cardboard bed. The power-supply cable was passed through a slit in the bed. The robot's ankles were tied to the bed. The robot was taught sit-up motions during ten sessions, as shown in Fig. 11(a). The states and actions were as described in Sec.III-A1. The robot learned the motions without a sub-goal reward, given by (5), and with a PTS-based sub-goal reward, given by (11). After 5 $runs$ for each case, the average convergences were compared. A run was organized as follows: if two consecutive $episodes$ were successful, the run was regarded as finished, but if the number of $episodes$ exceeded 20, the run was regarded as failed and terminated.

A reproduction of the learned motion is shown in Fig. 11(b). Without PTS, the number of $episodes$ of each run were $10, 7, 4, 11, 15$, (average 9.4), while those with PTS were $3, 3, 9, 6, 20$ (average 8.2). These results follow a similar trend to the simulation reported in Table III; the motor knowledge encoded as PTS enhanced the average convergence speed in terms of $episodes$ number.

B. Robustness of representation to interaction change

The robustness of the proposed motion feature was investigated by evaluating the similarity distributions between the reference motion and other motions (TS and SS motions). Three similarity indices were computed. The ED index distributions were influenced by the timing and amplitude gaps in the motions, while those of the TW index depend only on the amplitude gaps in the motions. The semantic differences of the motions could be observed by eliminating the timing and amplitude gaps. The evaluation was conducted on the motions described in Sec. III-A, and a reference motion was selected from the demonstrations.

1) Simulated distributions in sit-up motions: Sitting-up motions were evaluated as a first step. The comparison was based on three trials of learned SS sit-ups and the ten demonstrations of TS sit-ups obtained in Sec. III-A1. The similarity distributions between the i^r motion and other motions are plotted in Fig. 12(a). The horizontal axes denote the indices of similarity measurements between each motion and the reference motion. In the upper, central, and lower plots, these indices are based on ED given by (9), TW given by (10), and PTS given by (11), respectively. Motions that are more similar to the i^r motion lie further to the right in the plots, than motions that are less similar. The green vertical dotted-line indicates the motion obtained in the demonstration phase that is furthest from i^r. The vertical axis is used only for visual clarity; it spaces various types of motions. All points could be plotted on a horizontal line.

In the ED and the TW plots, the successful motions of the TS and SS are separated. The TS motions lie to the right of the boundary, while unsupported motions lie to the left. This situation is conceptualized in Fig. 1, which shows the clear separation of the two envelopes. Conversely, both types of successful motions are grouped together and are inseparable in the PTS plot.

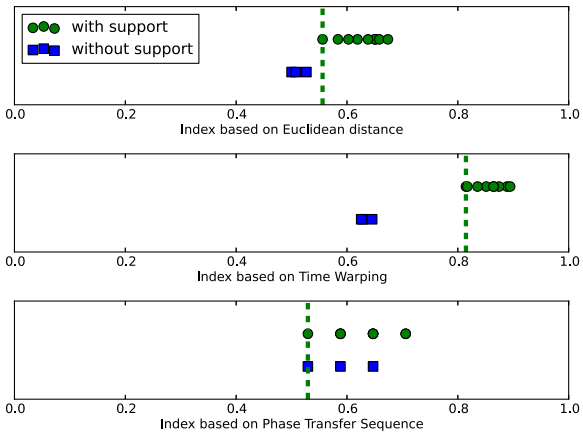
Knowledge extracted from the demonstration accelerates the learning process. In particular, similarity between the taught and current motions indicates a (partially) correct motion. However, the indices of similarity based on the ED and TW account for the details in the motion, and regard TS motions and SS motions as very different. Conceptually, using these indices, the knowledge of a reference motion might be confined to its allocated envelope (that of the TS motion, in our case), and unable to be exploited in motions allocated to other envelopes. Conversely, in PTS, SS motion learning is promoted by the knowledge of the TS motion. In fact, TS and SS motions can be considered similar if they possess common features, such as intermediate states that determine similar sequences of sensory changes in both motions.

Note that the TS and SS motions are more widely distributed in TW than in ED, because TW absorbs the timing gap, and therefore more precisely distinguishes the TS input sequences from the SS input sequences. In ED, whose distances depend on both timing and amplitude gap, the classification is less clear.

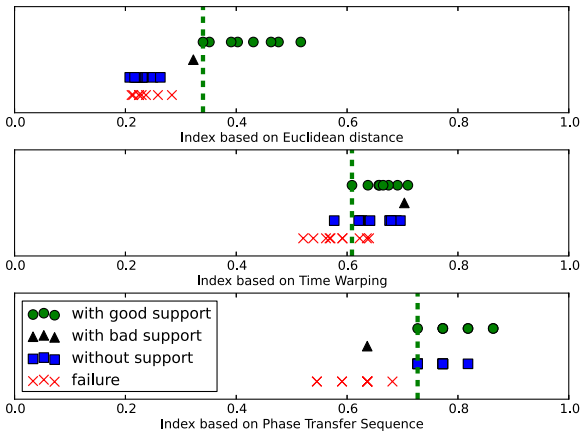
2) Simulated distributions in walking motions: Walking motions were evaluated as a second step. We produced ten successful and ten failed instances of SS walking by adding normally-distributed random noise ($N(0.0, 10.0)$) to the walking actions $A_w = \{a_0, a_1, a_2\}$ at time $\{0.0, 0.4, 0.8\}$ s. If the robot reaches the goal state s_g at time $t_e = 1.2$ s, the produced motions are regarded as successful. The comparison was based on 20 trials of the produced SS walks and the ten demonstrations of TS walks obtained in Sec. III-A2.

The experimental results are shown in Fig. 12(b). The TS walks are labeled "with good support" and "with bad support", while the SS walks and failed motions are labeled "without support" and "failure". In the PTS plot, the TS walking motions lying within the similarity boundary at 0.72 are labeled "with good support" and "without support", while failed trials and those labeled "with bad support" are excluded. The ability to separate the successful from failed cases is an important characteristic of PTS. Sensitivity to success and failure but robustness to interaction changes is a highly desirable property. Therefore, in terms of the PTS index, TS and SS walking motions are semantically identical, and can be classified as "walking".

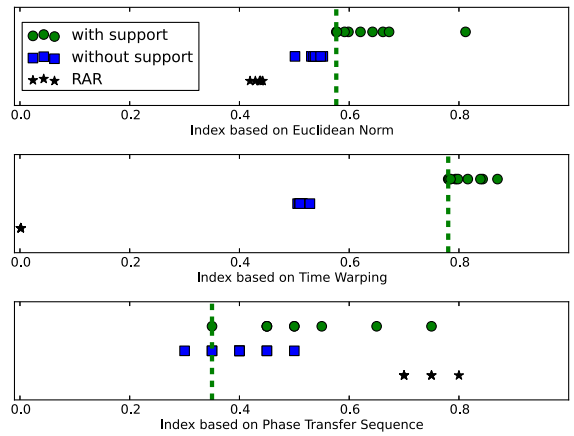
In the ED plot, the similarity of motions labeled "with good



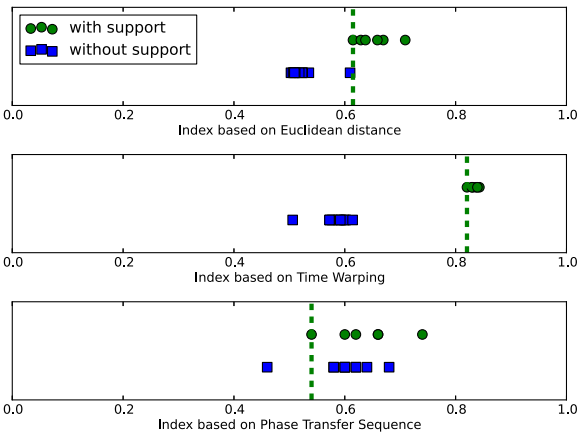
(a) Sit-up motions in simulation.



(b) Walk motions in simulation.



(c) Sit-up and roll-and-rise motions with a real robot.



(d) Walk motions with a real robot.

Fig. 12. Similarity distributions of TS and SS motions to the motion of the reference trial i^r (TS sitting-up in the sitting-up experiment and TS walking in the walking experiment). In each sub-figure, the vertical axes show the motion type, and the horizontal axes show the indices based on the ED (upper plot), TW (middle plot), and phase transfer sequence (lower plot). The green vertical dotted-line shows the boundary of the TS motion; if the motion similarity lies to the right of this boundary, the motion is considered to belong to the reference motion. In all sub-figures except (b), the similarities of the TS and SS motions are denoted as "with support" and "without support," respectively. In these sub-figures, the SS motions are inside the boundary in the PTS plot, but lie outside the boundary in the other plots. In (b), the successful trials of TS walking (SS walk) are labeled "with good support" and "with bad support" ("without support"), while failure trials are labeled "failure." In this sub-figure, the PTS plot shows that SS motions are inside the boundary of the TS motion (except "with bad support"), while failed motions lie outside the boundary. The walking motions in the TW plot are almost exclusively confined within the boundary, but some failed motions also appear within the boundary. In the ED plot, although failed motions appear outside the boundary, SS motions are also excluded. In (c), the successful trials of the roll-and-rise motion are denoted by "RAR". A preliminary analysis of RAR motion is given in Sec. IV-3.

support" exceeds 0.34, while failed motions and SS motions are both concentrated below the boundary, at a similarity index around 0.24. Therefore, in the ED representation, SS walking acquired from the knowledge of TS walking might be compromised by the separation of the envelopes containing the two types of motion. Moreover, SS walking might be grouped among the failures.

In the TW plot, motions with similarity indices exceeding the 0.61 boundary include failures as well as successful walks. In the evaluated settings, the force applied to the robot during its walking sessions and the impact of falling in failed cases is impulsive compared to the motion time. Consequently, the amplitude gaps between the motions are rather small. The separation of TS and SS walking by ED but not by TW can be explained by the rhythmic nature of walking motions. Because it absorbs the timing gap, TW places motions of different periods in the same group, while ED regards them as different.

Figure 10(c) illustrates an incorrect teaching trial, labeled "with bad support" in Fig. 12(b). In this case, although the overall trial was successful, the robot stumbled on the 3rd step, and the motion phase transitions differed from the reference PTS. Such incorrect phase transitions are probable if the experimenter is not adequately trained.

3) Sit-up and walk distributions in a real robot: Next, sitting-up and walking experiments were conducted on a real robot. The evaluation involved ten trials of TS sit-ups and ten trials of the SS sit-ups described in Sec. III-A3. During the walking experiments, the robot undertook seven walking sessions, with the experimenters holding its arms, from which a reference PTS $\mathbf{o}_r$ was selected. The robot was allowed ten reproductions of the SS walk obtained in the simulation. To steady the robot's motion, its torso was held to compensate for the torque of the roll joints during walking without compromising other features of the motion. The walking motion

(a) Interaction for teaching the walk motion.

(b) Reproduction of learned walk motion.

Fig. 13. Walk experiments by the real robot.

was self-generated by the robot.

The similarity distributions in the sitting-up and walking motions are plotted in Fig. 12(c) and 12(d). Figures 13(a) and 13(b) show the interaction during walk teaching and a reproduction of the learned walking motion, respectively. The results are consistent with those of the simulation. In the PTS plot, both TS and SS motions are distributed within the boundaries of the TS motion (over 0.35 for sitting-up and over 0.53 for walking), while the SS motions are excluded from the boundaries of the TW and ED plots. These results reveal that PTS provides knowledge that can be later utilized in similar motions under different physical interactions.

The walking results in the TW plot differ from their simulated results, because amplitudes exert a stronger effect in real robot experiments than in simulations. In the simulations, the robot received discrete support by the impulsive force computed from the mouse drag-and-drop movement, while the actual robot was continuously supported throughout the demonstration. This dynamical difference probably explains the enhanced amplitude gap in the real experiments.

IV. Conclusion and Discussion

In this study, we proposed a motion feature (PTS) that encodes phase transitions in multiple time sequences. We applied the feature to the LfD of the whole body motions of a robot. PTS was shown to robustly characterize the motion dynamics, in terms of timing and amplitude of the sequence profiles. This property allows PTS to evaluate motions executed in different physical interactions as similar, provided that the motions possess certain similar features. The PTS were extracted by the system from human taught motions in TS interactions, and were used to guide the RL of motions in SS interactions. In simulations and in experiments involving a real humanoid robot, the similarity distributions of the motions in different interactions suggested that PTS is robust to changes in physical interactions. Moreover, under PTS, the robot could exploit previous knowledge to accelerate its learning convergence in both sitting-up and walking motions.

1) Future works: The achievements reported in this work may contribute to the LfD development, since PTS may complement other approaches such as DMP. In DMP, representations of a particular interaction can be re-represented in another interaction merely by reproducing the motion in the new interaction. Figure 1 shows that the motions in different interactions are separated. Provided that different interactions yield successful motions, the envelopes encompassing these interactions can be mapped onto each other. For this purpose, our method can apply knowledge gained from previous motions to speedily learn new motions.

Our present method focused on extracting sub-goals from reference data; therefore, it adopted discrete RL as a first step. A more adaptable version of PTS could be achieved by rendering it compatible with continuous RL frameworks. In particular, the PI^2 algorithm [18] is a learning method based on a reference trajectory in a high-dimensional system. This algorithm, which accepts user-input sub-goals, has been shown to achieve accurate tracking through via-points either in joint space or end-effector space, and manipulation of tasks (including physical interactions with the environment) by adjusting the impedance of the end-effector. In this setup, the sub-goals in PTS could be more abstractly represented, and rendered more adaptable to changes in physical interactions.

The purpose of the hierarchical RL framework [19] in continuous space was to achieve standing-up behavior at a practical learning speed. The system has two layers. In the first layer, Q(λ), the agent learns to achieve final goals and in the second layer, sub-goals are achieved by actual motion generated from a continuous actor-critic method. Also in this scenario, PTS could supply abstract sub-goals in the first layer, enabling adaptation to physical interaction changes.

Our methodology should also be extendible to multiple robots with various sensors and undertaking different roles. In fact, our system is based on generic approaches such as SVD and LCS. SVD is applicable to any form of time sequences, while LCS is applied in several learning systems, such as robot manipulation [10], task learning [20], and skill-transfer with haptic devices [21]. More sophisticated tasks such as dancing [5] are also suitably accommodated by our method; for example, the robot could apply knowledge gained from TS dancing to learn SS dancing. Since dancing can be segmented into key postures, and postural changes are detectable by SST, sub-goals could be obtained that accelerate the learning.

However, our approach would benefit from further improvement. Although our representation is robust to changes in physical interactions, it degrades features of the motion such as smoothness in the trajectories and velocities of the joint angles. Combination with DMP [3] is expected to alleviate these problems. Moreover, posture control for maintaining locomotive balance, based on ZMP [22] or reflex-based control [23], should be integrated into the method. Walking combines two basic controls: locomotive control and balancing the motion. As a first step, we have focused on the abstracted features that allow the robot to move forward under different physical interaction conditions. In future work, we will introduce and evaluate automatic balance control in the method.

2) Preliminary sit-up skill analysis: To reveal the sub-goals of sitting-up motions, we investigated the LCS between TS sit-up of i^{τ} and SS sit-ups. From the observed set of LCS, typical sub-goals were empirically determined as $''\mathbf{b}, \mathbf{54}, \mathbf{1}, \mathbf{67}, \mathbf{2}, \mathbf{a}''$, where the head gyro $\mathbf{b}$, hip joint encoders $\mathbf{54}$, right arm

Fig. 14. Demonstration of roll-and-rise motion by the real robot.

force $\mathbf{1}$, shoulder joint encoders $\mathbf{67}$, left leg force $\mathbf{2}$, and head angle $\mathbf{a}$ are changed in this order. Thus, even discrete motions such as sitting up are achieved via a series of sub-goals. The sub-goals can be theoretically determined using the MLCS algorithm [24], which identifies the LCS in multiple strings.

3) Roll-and-rise analysis in a real robot: To evaluate the applicable region of the proposed representation, we also conducted a real robot experiment involving RAR motion. which is identical to the sitting-up motion, but a more complex task. As the complexity of the task increases, the number of phase transitions is expected to increase. During the RAR experiment, the robot was laid on a bed similar to that used in the sitting-up experiments, but equipped with a step for elevating the robot body, as shown in the leftmost section of Fig. 14. The seven RAR actions $A_r = \{a_0, ..., a_6\}$ are listed in Table II. To ensure safety of the real robot, the switching times of the actions, from a_0 to a_6, were adjusted by the human experimenter. A successful RAR trial is shown in Fig. 14. The PTS generated from six RAR trials was compared with that of the TS sitting-up motion of i^r.

The experimental results are plotted in Fig. 12(c). In the PTS plot, the RAR distribution (minimized at 0.7) is well within the boundary of the TS sitting-up motion (0.35), while the ED- and TW-based RAR distributions lie outside their boundary. In the Euclidean plot, the RAR distribution is separated from the distributions of the TS and SS sitting-up motions. In the TW plot, the similarities of RAR are approximately 0 because a huge amplitude gap exists between the RAR and sitting-up motions. However, we may reasonably question why RAR is more similar to the reference motion than TS (SS) sitting-up.

To clarify this effect, we investigated the number of phase transitions in each task motion, and present the results in Table IV. The average lengths of RAR and TS sit-up (SS sit-up) in PTS were 45.67 and 35.39 (29.0), respectively. If more phase transitions occur in dexterous tasks than in the reference motion, the PTS similarity of (11) receives a higher score, because the consequent longer sequences will more likely contain sub-sequences common to the reference sequence than shorter sequences. For instance, a noisy motion that randomly generates phase transitions at each sensor over a long period may yield a similarity close to 1. Since RAR is a dexterous task, it probably generates a longer string of phase transitions than sitting-up motions; therefore, acquires a higher similarity score. Therefore, dexterous motions that generate longer sequences than the reference sequence may impede learning enhancement, since the system lacks information on noisy or rapidly changing motions. However, when the sequence of the SS motion is shorter than the reference sequence of the TS motion, our approach is a valuable learning guide, as verified throughout this study.

TABLE IV
SUMMARY OF PHASE TRANSITIONS OF THE TASK MOTIONS.

task	situp			walk	
	TS	SS	RAR	TS	SS
average	35.39	29.0	45.67	50.7	49.0
variance	6.63	7.4	12.22	19.9	43.0

APPENDIX A

Algorithm 1 The Generation of Phase Transfer Sequence

INPUT :

$X(t) = \{\mathbf{x}_1(t), ..., \mathbf{x}_{N_p}(t)\}$; multiple time sequences of multiple sensors.

$\mathbf{x}_i(t) = \{x_i(t_s), ..., x_i(t)\}$; the sequence of the sensor p_i from the start time t_s to time t.

N_p ; the number of the sensors.

$P = \{p_1, ..., p_{N_p}\}$; a set of sensors.

$L^P = \{l_1^P, ..., l_{N_p}^P\}$; a set of symbols for each sensor.

OUTPUT :

$\mathbf{o}$; Phase Transfer Sequence

Initialize $\mathbf{o} = \emptyset$ as empty symbol.
(*Binarization*)
for $i = 1$ to N_p **do**
  The change score sequence $\widetilde{\mathbf{x}}_i(t) = \{\widetilde{x}_i(t_s), ..., \widetilde{x}_i(t)\}$ of $\mathbf{x}_i(t)$ are computed using RSST by Algorithm 2.
  for all $\widetilde{x}_i(\tilde{t}) \in \widetilde{\mathbf{x}}_i(t)$ $(t_s \leq \tilde{t} \leq t)$ **do**
    If $\widetilde{x}_i(\tilde{t}) \neq 1$ then $\widetilde{x}_i(\tilde{t}) \leftarrow 0$.
  end for
end for
(*Symbolization*)
for $i = 1$ to N_p **do**
  The changing times $\mathbf{t}_i = \{t_i^1, ..., t_i^{N_i^c}\}$ are collected, where N_i^c is the number of the peaks of the change scores at which $\widetilde{x}_i(t_i^j) = 1$ $(1 \leq j \leq N_i^c)$.
  Each time is labeled with same symbol l_i^P corresponding to the sensor p_i.
end for
Obtain the PTS $\mathbf{o}$ by concatenating the symbols in temporal order.

APPENDIX B

Algorithm 2 Robust Singular Spectrum Transformation

INPUT :

$\mathbf{x}(t)$; time sequence from the start time t_s to the time t.

n_r, n_c ; the row and column size of Hankel Matrix.

OUTPUT :

$\widetilde{\mathbf{x}}(t)$; the sequence of the final changing score.

OTHER VARIABLES :

$x(\tilde{t})$; a point of $\mathbf{x}(t)$ at time $\tilde{t}$ $(t_s \leq \tilde{t} \leq t)$.

$H(\tilde{t}) = [seq(\tilde{t} - n_c),\ ...,\ seq(\tilde{t} - 1)]$; Hankel Matrix.

$seq(\tilde{t}) = \{x(\tilde{t} - n_r + 1),\ ...,\ x(\tilde{t})\}^T$; the sequence with length n_r.

(Goes to next page)

(From previous page)

1: **for** $\tilde{t} = t_s$ to t **do**

2: Set $H_p(\tilde{t}) = [seq(\tilde{t}-n_c), ..., seq(\tilde{t}-1)]$ as past matrix, $H_f(\tilde{t}) = [seq(\tilde{t}+1), ..., seq(\tilde{t}+n_c)]$ as future matrix.

3: Find the past and future features of $H_p(\tilde{t})$ and $H_f(\tilde{t})$ by singular value decomposition:

$$H(\tilde{t}) = U(\tilde{t})S(\tilde{t})V(\tilde{t})^T. \quad (13)$$

In order to get the essential features of the sequence, calculate the number of left singular vectors $l(\tilde{t})$ of past and future patterns ($l_p(\tilde{t})$ and $l_f(\tilde{t})$) as follows: sort the singular values of $H(\tilde{t})$, find where the tangent of their accumulated sum has an angle below $-\pi/4$.

4: Project future singular vectors $\chi_i(\tilde{t})$ $(i \leq l_f(\tilde{t}))$ onto the hyper plane built by the past singular vectors $U_{l_p(\tilde{t})}$:

$$\zeta_i(\tilde{t}) = \frac{U^T_{l_p(\tilde{t})}\chi_i(\tilde{t})}{\|U^T_{l_p(\tilde{t})}\chi_i(\tilde{t})\|}(i \leq l_f(\tilde{t})). \quad (14)$$

The norm of each projection vector $\zeta_i(\tilde{t})$ represents the difference between each $\chi_i(\tilde{t})$ and the hyper plane. Next calculate the change score by $cs_i(\tilde{t}) = 1 - \|\zeta_i(\tilde{t})\|$. If the $\chi_i(\tilde{t})$ is on the hyper plane ($\|\zeta_i(\tilde{t})\| = 1$), $cs_i(\tilde{t})$ become 0; the future and the past features are similar.

5: Calculate the preliminary change score by

$$\widehat{x}(\tilde{t}) = \frac{\sum_{k=1}^{l_f(\tilde{t})} \lambda_i(\tilde{t}) cs_i(\tilde{t})}{\sum_{k=1}^{l_f(\tilde{t})} \lambda_i(\tilde{t})}, \quad (15)$$

where $\lambda_i(\tilde{t})$ are the eigenvalues of the future feature matrix $H_f(\tilde{t})$.

6: **end for**

7: **for** $\tilde{t} = t_s$ to t **do**

8: In order to filter the noise, update $\widehat{x}(\tilde{t})$ by:

$$\widetilde{x}(\tilde{t}) = \widehat{x}(\tilde{t}) \times \|\mu_f - \mu_p\| \times \|\sigma_f - \sigma_p\|, \quad (16)$$

where μ_p (μ_f) and σ_p (σ_f) are the mean and variance of a past (future) sequence of length n_r at $\widehat{x}(\tilde{t})$.

9: **end for**

10: Normalize $\widetilde{\mathbf{x}}(\tilde{t})$ with its local maximum.

11: Get the sequence of the final changing score $\widetilde{\mathbf{x}}(t)$.

ACKNOWLEDGMENT

The authors would like to thank Stefano Saliceti, Julien Jenvrin and Marco Randazzo for his helps in experiments with a real robot. This work is supported by JSPS KAKENHI Grant Number 2457062, JSPS Core-to-Core Program A. Advanced Research Networks, EU FP7 project CHRIS (Cooperative Human Robot Interaction Systems FP7 215805) and EU FP7 project Xperience (Robots Bootstrapped through Learning and Experience, FP7 97459).

REFERENCES

[1] P. Zelazo, N. Zelazo, and S. Kolb, "Walking in the Newborn," *Science*, vol. 176, no. 4032, p. 314, 1972.

[2] D. A. Ulrich, B. D. Ulrich, R. M. Angulo-Kinzler, and J. Yun, "Treadmill training of infants with Down syndrome: evidence-based developmental outcomes," *PEDIATRICS*, vol. 108, no. 5, pp. e84–e84, 2001.

[3] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," *Advances in Neural Information Processing Systems*, vol. 15, pp. 1523–1530, 2002.

[4] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, "Learning from demonstration and adaptation of biped locomotion," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 79–91, 2004.

[5] S. Nakaoka, A. Nakazawa, K. Yokoi, H. Hirukawa, and K. Ikeuchi, "Generating whole body motions for a biped humanoid robot from captured human dances," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2003, pp. 3905–3910.

[6] Y. Kuniyoshi, Y. Ohmura, K. Terada, A. Nagakubo, S. Eitoku, and T. Yamamoto, "Embodied basis of invariant features in execution and perception of whole-body dynamic actions–knacks and focuses of Roll-and-Rise motion," *Robotics and Autonomous Systems*, vol. 48, no. 4, pp. 189–201, 2004.

[7] S. Ikemoto, H. Amor, T. Minato, B. Jung, and H. Ishiguro, "Physical Human-Robot Interaction: Mutual Learning and Adaptation," *IEEE Robot. Automat. Mag.*, vol. 19, no. 4, pp. 24–35, Feb. 2012.

[8] K. Kuwayama, S. Kato, T. Kunitachi, and H. Itoh, "Motion control for humanoid robots based on the motion phase decision tree learning," in *Proceedings of the 2004 International Symposium on Micro-Nanomechatronics and Human Science, 2004 and The Fourth Symposium Micro-Nanomechatronics for Information-Based Society, 2004.*, 2004, pp. 157–162.

[9] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Trans. Syst., Man, Cybern. B*, vol. 37, no. 2, pp. 286–298, 2007.

[10] V. Kruger, D. Herzog, S. Baby, A. Ude, and D. Kragic, "Learning Actions from Observations," *Robotics & Automation Magazine, IEEE*, vol. 17, no. 2, pp. 30–43, 2010.

[11] T. Idé and K. Inoue, "Knowledge discovery from heterogeneous dynamic systems using change-point correlations," in *Proc. SIAM Intl. Conf. Data Mining*, 2005, pp. 571–575.

[12] L. Bergroth, H. Hakonen, and T. Raita, "A survey of longest common subsequence algorithms," in *Seventh International Symposium on String Processing and Information Retrieval, 2000. SPIRE 2000. Proceedings.*, 2000, pp. 39–48.

[13] Y. Mohammad and T. Nishida, "Robust singular spectrum transform," *Next-Generation Applied Intelligence*, pp. 123–132, 2009.

[14] C. Watkins and P. Dayan, "Technical note: Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[15] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The iCub humanoid robot: an open platform for research in embodied cognition," in *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, 2008, pp. 50–56.

[16] G. Metta, P. Fitzpatrick, and L. Natale, "Yarp: Yet another robot platform," *International Journal on Advanced Robotics Systems*, vol. 3, no. 1, pp. 43–48, 2006.

[17] K. Hosoda and Y. Ishii, "External rotation as morphological bootstrapping for emergence of biped walking," in *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*, 2010, pp. 317–322.

[18] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 820–833, 2011.

[19] J. Morimoto and K. Doya, "Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning," *Robotics and Autonomous Systems*, 2001.

[20] M. N. Nicolescu and M. J. Mataric, "Natural methods for robot task learning: Instructive demonstrations, generalization and practice," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. ACM, 2003, pp. 241–248.

[21] C. H. Park, J. W. Yoo, and A. M. Howard, "Transfer of skills between human operators through haptic training with robot coordination," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 229–235.

[22] M. VukobratoviĆ, B. Borovac, and V. Potkonjak, "ZMP: A review of some basic misunderstandings," *International Journal of Humanoid Robotics*, vol. 3, no. 2, pp. 153–176, 2006.

[23] Q. Huang and Y. Nakamura, "Sensory reflex control for humanoid walking," *IEEE Trans. Robot.*, vol. 21, no. 5, pp. 977–984, 2005.

[24] Q. Wang, D. Korkin, and Y. Shang, "A Fast Multiple Longest Common Subsequence (MLCS) Algorithm," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 23, no. 3, pp. 321–334, 2011.

Toshihiko Shimizu attained his B.Eng., M.Eng. and D.Eng. degrees from Osaka university, Osaka, Japan, in 2007, 2009 and 2013. He is now an assistant professor in the Department of Mechanical Engineering, Kobe City College of Technology. He became a researcher at the Istituto Italiano di Tecnologia, Genova, Italy, in 2010. He had been a JSPS Research Fellow from April 2012 to September 2013. His research interests include machine learning with skill transfer, biologically inspired control and physical human-robot interaction.

Ryo Saegusa has been a project associate professor with the Center for Human-Robot Symbiosis Research, Toyohashi University of Technology since 2012. He received his B.Eng., M.Eng. and D.Eng. degrees in Applied Physics from Waseda University in 1999, 2001 and 2005. He was a research associate with the Department of Applied Physics, Waseda University from 2004 to 2007 and moved to the Robotics, Brain and Cognitive Sciences Department at the Istituto Italiano di Tecnologia as a researcher from 2007 to 2012. His research interests include machine learning, computer vision, signal processing, cognitive robotics, and health care robotics.

Shuhei Ikemoto attained his Ph.D. in engineering from Osaka University in March 2010. He was a JSPS Research Fellow from April 2009 to March 2010. He is now an assistant professor in the Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University. His research interests include machine learning, biologically inspired algorithms and physical human-robot interaction.

Hiroshi Ishiguro attained his D.Eng. degree from Osaka University, Osaka, Japan, in 1991. In 1991, he started working as a Research Assistant in the Department of Electrical Engineering and Computer Science, Yamanashi University, Yamanashi, Japan. Then he moved to the Department of Systems Engineering, Osaka University, as a Research Assistant in 1992. In 1994, he became an Associate Professor, Department of Information Science, Kyoto University, Kyoto, Japan, and started research on distributed vision using omnidirectional cameras. From 1998 to 1999, he worked in the Department of Electrical and Computer Engineering, University of California, San Diego, as a Visiting Scholar. In 2000, he moved to the Department of Computer and Communication Sciences, University, Wakayama, Japan, as an Associate Professor and became a Professor in 2001. He moved to the Department of Adaptive Machine Systems, Osaka University in 2002. He is now a Professor in the Department of Systems Innovation, Osaka University, and a Group Leader at ATR Intelligent Robotics and Communication Laboratories, Kyoto. Since 1999, he has also been a Visiting Researcher at ATR Media Information Science Laboratories, Kyoto.

Giorgio Metta is a director of the iCub Facility at the Istituto Italiano di Tecnologia (IIT) where he coordinates the development of the iCub robotic platform/project. He holds a MSc cum laude (1994) and PhD (2000) in electronic engineering both from the University of Genoa. From 2001 to 2002 he was a postdoctoral associate at the MIT AI-Lab. He was previously with the University of Genoa and since 2012 a professor of Cognitive Robotics at the University of Plymouth (UK). He is a deputy director of IIT delegate to the international relations and external funding. In this role he is a member of the board of directors of euRobotics aisbl, the European reference organization for robotics research. His research activities are in the fields of biologically motivated and humanoid robotics and, in particular, in developing humanoid robots that can adapt and learn from experience. He is an author of approximately 200 scientific publications. He has been working as a principal investigator and research scientist in about a dozen international as well as national funded projects.